



UNITED STATES PATENT AND TRADEMARK OFFICE

PH
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/645,740	08/21/2003	Eric L. Barsness	ROC920030203US1	9120
30206	7590	02/21/2006		EXAMINER
IBM CORPORATION				SYED, FARHAN M
ROCHESTER IP LAW DEPT. 917				
3605 HIGHWAY 52 NORTH			ART UNIT	PAPER NUMBER
ROCHESTER, MN 55901-7829			2165	

DATE MAILED: 02/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/645,740	BARSNESS ET AL.
	Examiner Farhan M. Syed	Art Unit 2165

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 08 February 2006.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-42 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-42 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date <u>20060208</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. Claims 1-42 are pending.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-11, 13-31, and 33-42 are rejected under 35 U.S.C. 102(b) as being anticipated by Risch (U.S. Patent No. 5,471,629).

As per claims 1, 21, and 41, Risch teaches a method for optimizing execution of a query that is repeatedly issued from an issuing entity against a database (i.e. “*A method of monitoring objects in a database system.*” The previous text clearly indicates that monitoring objects is an execution of a query that is repeatedly issued from an issuing entity.)(Abstract), the method comprising: executing the query against the database to obtain an initial query result (i.e. “*A database monitor can be compiled once for a given query, as discussed in the aforesaid patent and in Risch, “Monitoring Database Objects” in Proceedings of the 15th Conference on Very Large Databases (Amsterdam, Holland) 1989, pp. 445-453.*” The preceding text clearly indicates that an initial query result is the given query.)(column 5, lines 64-67; column 6, line 1); returning the initial query result to the issuing entity (i.e. “*the tracker can freely access the database to retrieve the current result of the monitored query.*” The preceding text clearly indicates that the monitored query is the initial query result.

Any ordinary person skilled in the art knows when someone executes a query, results are returned back to the user.)(column 4, lines 37-38); and managing re-execution of the query by: storing the initial query result in a temporary query result data structure (i.e. *"The system creates the Function Change table (block 202) and then proceeds with an update task as directed by the client (block 203.)"*) The preceding text clearly illustrates that a temporary query result data structure is a function change table and the system is the re-execution of the query.)(column 8, lines 6-8); executing a trigger procedure (i.e. *"Systems of "triggers" have been proposed for relational database systems; such triggers typically invoke a database procedure upon updates of user-specified base relations."*)(column 2, line 65; column 3, line 1) for tracking database changes (i.e. *"The invoked procedure is sometimes referred to herein as the "tracking procedure" ("tracker" for short) because it "tracks"--that is, is invoked by a change in--the monitored attribute."*)(column 4, lines 1-3); when the trigger procedure detects a database change, updating the temporary query result data structure on the basis of the detected database change (i.e. *"Determining whether the criterion for notification has been satisfied and, if so, notifying the appropriate clients, is preferably accomplished by means of the Check Monitors procedure as illustrated in FIG. 3. As described above, the procedure is initiated (block 301) by a procedure call generated during an update session (the procedure may also be called directly by a client as will be discussed in more detail in a subsequent paragraph). Any update function call which might have resulted in a change to a monitored attribute value is detected (block 302). In most instances this is accomplished by reference to the Function Change table and to a "Function Dependence" table (to be described hereafter)."* *"Because in this instance the procedure was called by a commit request made by an update client, the update client is the one which receives the notification. The Function Change table is cleared (block 307) and the procedure ends (block 308.)"* The preceding text clearly indicates that a Check Monitors procedure is a type of a trigger procedure that updates the temporary query data result, which is the Function Change table and further updates the client request, which is the initial query.)(column 8, lines 15-25 and 36-40); and when the query is re-executed

against the database, returning the temporary query result data structure to the issuing entity as query result (i.e. *"Then the saved change notification is sent to any other client which had requested monitoring of the changed attribute value (block 209). If there is another update task to perform (block 210), such task is performed (block 203) and the above-described steps are repeated."* The preceding text clearly indicates that the saved change notification is the new results in the temporary query result data structure that updated the initial query request by the client and any other client which requested monitoring is the query re-execution.)(column 8, lines 49-54).

As per claims 2, 17, 22, and 37, Risch teaches a method wherein managing the re-execution of the query is only performed if the query includes a flag indicating repeated execution (i.e. *"A "retrieve always" mechanism in another system causes queries to be re-executed upon each update to specified relations (M. Stonebraker, "Triggers and Inference in Database Systems," in M. Brodie and J. Mylopoulos (eds.), On Knowledge Based Management Systems, Springer-Verlag, 1986)." The preceding text clearly indicates that a flag is a "retrieve always" mechanism.*)(column 2, lines 60-65).

As per claims 3, 18, 23, and 38, Risch teaches a method further comprising, prior to executing the query against the database to obtain the initial query result: determining an amount of system resources required for executing the query (i.e. *"It will be apparent that the comparison can be more complex than the above examples, as in "fixed thresholding" or "moving averages". Fixed thresholding can be specified by inequality comparisons. An example of time-consuming reasoning that may be required in making a comparison is described in Washington et al., "Input Data in Real-Time AI Systems" 11th International Joint Conference on Artificial Intelligence, 1989, pages 250-255. Simple comparisons are more easily optimized than complex ones and ordinarily impose less overhead on the system."* The preceding text clearly indicates that determining

an amount of system resource required to execute the query is the overhead on the system.)(column 11, lines 31-40); and managing the re-execution of the query only if the determined amount exceeds a predetermined threshold value (i.e. *"Whether the criterion has been satisfied is determined by determining whether the monitored attribute may have been affected by the update transaction and, if so, determining whether the value of the attribute has changed by an amount which exceeds the minimum change value."* The preceding text clearly indicates that a predetermined threshold value is the minimum change value.)(column 5, lines 60-64).

As per claims 4, 19, 24, and 39, Risch teaches a method wherein the predetermined threshold value is one of user-specific, application-specific and system-specific (i.e. *"If a monitored attribute in a database is being intensively updated, the reactivity can become very high, resulting in high overhead and network traffic. In the worst case, the monitoring application will spend all its time invoking tracking procedures in response to monitor notifications. Tuning the monitor according to the method of the invention solves this problem. The change significance parameter provides one way to reduce the reactivity of a monitor activation. This parameter has the effect of making the tracker react only to "significant" changes in the value of the monitored attribute. The client defines what is a "significant" change by specifying a minimum value for the change significance parameter. This minimum is specified either as an absolute difference (for example, ten dollars) or as a relative difference (for example, a change of 10%). For a query that returns a set of values, the significance test is preferably applied to every value in the set such that if any value changes by an amount that exceeds the minimum change value, the client is notified. The client may dynamically change the value of this parameter as needed. For example, it may be desired to decrease the reactivity temporarily if the client must concentrate on a critical task."* The preceding text clearly indicates that the predetermined threshold value is the significant change in value. It is user-specific, where the client may dynamically change the value of this parameter as needed, application-specific, where the threshold

value is needed to reduce high overhead, and system-specific, where the threshold value is needed to reduce network traffic is system-specific.)(column 11, lines 8-30).

As per claims 5, 15, 25, and 35, Risch teaches a method further comprising: providing a notification to the issuing entity indicating the updating (i.e. *"Notifying a client preferably comprises interrupting any task then being performed by the client and invoking a predesignated tracking procedure, for example by sending the notification to a portion of the client program referred to as a "monitor server". If the client being notified is the one which called the Check Monitors procedure (usually this will be the client which requests that updates be committed), this is done when a change in the monitored attribute value is detected (block 309 of FIG. 3)." The preceding text clearly indicates that the issuing entity is the client.)*(column 9, lines 2-10).

As per claims 6 and 26, Risch teaches a method wherein the trigger procedure executes at least one of: a delete trigger for tracking delete operations performed on the database; an insert trigger for tracking insert operations performed on the database; and an update trigger for tracking update operations performed on the database (i.e. *"Determining whether the criterion for notification has been satisfied and, if so, notifying the appropriate clients, is preferably accomplished by means of the Check Monitors procedure as illustrated in FIG. 3. As described above, the procedure is initiated (block 301) by a procedure call generated during an update session (the procedure may also be called directly by a client as will be discussed in more detail in a subsequent paragraph). Any update function call which might have resulted in a change to a monitored attribute value is detected (block 302)." The preceding text clearly indicates that the trigger procedure executes a check monitor procedure, which is a form of an update trigger.)*(column 8, lines 15-24).

As per claims 7 and 27, Risch teaches a method wherein managing the re-execution of the query comprises: determining a timeframe from a predefined time parameter (i.e. *“Activating a monitor in an application process informs the database system that it must henceforth invoke the tracker whenever a result change is detected for a given query and parameters.”* “*... the criterion is a time delay, that is, a requirement that a minimum amount of time have elapsed subsequent to a previous event such as a previous change in the value of the monitored attribute...*” The preceding text clearly indicates that a timeframe is a time delay and a predefined time parameter is a minimum amount of time has elapsed.)(column 5, lines 2-5; column 12, lines 54-57); and managing the re-execution in accordance with the determined timeframe (i.e. *“...and if the minimum time has not elapsed as indicated by a “no” output from a “minimum time elapsed?” block 317, the procedure ends without notification.”* The preceding text clearly indicates that a re-execution takes place according to the determined timeframe.)(column 57-60).

As per claims 8 and 28, Risch teaches a method wherein determining the timeframe comprises: extracting the predefined time parameter from the query (i.e. *“Activating a monitor in an application process informs the database system that it must henceforth invoke the tracker whenever a result change is detected for a given query and parameters.”* The preceding text clearly indicates that a query possess certain parameters. *“Another way to decrease reactivity is by means of the tracking delay time parameter.”* The preceding text clearly indicates that a predefined time parameter is a tracking delay time parameter that is extracted from the query.)(column 5, lines 2-5; column 11, lines 41-42).

As per claims 9 and 29, Risch teaches a method wherein the predefined time parameter indicates a point of time when execution of the trigger procedure is to be

terminated (i.e. *"A monitor is deactivated either explicitly or when the process is terminated."* *"The invoked procedure ends at time T10 and the system continues by clearing the Function Change table, committing the data, and notifying the remote client's monitor server, and committing the data."* The preceding text clearly indicates that a predefined time parameter is time T10, the invoked procedure is the trigger procedure, and end is terminated.)(column 5, lines 4-5; column 14, lines 30-33).

As per claims 10 and 30, Risch teaches a method further comprising, after the determined timeframe is elapsed: deleting the temporary query result data structure (i.e. *"However, for purposes of the example it is assumed that there are no further changes and that the update client ends the update session at time T15. The system deletes the Function Change table for that client and removes the client from the Client Address and Attribute Value tables."* The preceding text clearly indicates that the elapsed timeframe is time T15 and the temporary query result data structure is the function change table.)(column 14, lines 58-60); and if the query is re-executed against the database, executing the query against the database to obtain a new initial query result (i.e. *"Thereafter, beginning at time T4 the update client continues updating and the system updates the Function Change table (see block 205 of FIG. 2) while it performs the tasks as commanded. At time T5 the remote client begins using the system. If the remote client were to initiate an update session, another Function Change table would be created..."* The preceding text clearly indicates that after the query is updated with the new results, which updating the system and function change tables, if a remote client initiates an update session, which is based on re-executing the query, a new function change table is created, based upon the new results.)(column 14, lines 8-10).

As per claims 11 and 31, Risch teaches a method wherein the predefined time parameter indicates a point of time when the temporary query result data structure is to be deleted (i.e. *"The invoked procedure ends at time T10 and the system continues by clearing the*

Function Change table, committing the data, and notifying the remote client's monitor server, and committing the data." The preceding text clearly indicates that a predefined time parameter is the time T10, clearing is deleting, and the temporary query result data structure is the function change table.)(column 14, lines 30-33).

As per claims 13 and 33, Risch teaches a method wherein the predefined time parameter is one of a query-specific parameter, a database-specific parameter, an issuing entity-specific parameter and a system-specific parameter (i.e. "*In another aspect of the invention, if a client requests notification of any changes which have occurred in an attribute then being monitored for said client, the system determines whether a predetermined criterion respecting a change in the monitored attribute has been satisfied and, if the criterion has been satisfied, the system notifies said client accordingly. This notification invokes the tracker of that client. The criterion is any of four tuning parameters. These parameters include a change significance parameter, a tracking delay time parameter, a nervousness parameter, and a synchronous initiation parameter.*" The preceding text clearly indicates that a change significant parameter, a tracking delay time parameter, a nervousness parameter, and a synchronous initiation parameter are all types of predefined time parameters, which includes query-specific parameter, database-specific parameter, issuing entity-specific parameter, and system-specific parameter, and predefined timing parameters are forms of tuning parameters.)(column 5, lines 45-57).

As per claims 14, 34, and 42, Risch teaches a method for optimizing system resource use for execution of a query that is repeatedly issued from an issuing entity against a database in a data processing system, the method comprising: tracking changes to the database (i.e. "*The invoked procedure is sometimes referred to herein as the "tracking procedure" ("tracker" for short) because it "tracks"--that is, is invoked by a change in--the monitored attribute.*") (column 4, lines 1-3); and if a database change is detected: retrieving a

temporary query result data structure storing a query result that was previously obtained in response to execution of the query (i.e. *"The record of database update transactions is preferably kept by creating and updating a "Function Change" table during an "Update Session" procedure as illustrated in FIG. 2."* The preceding text clearly indicates that a temporary query result data structure is the function change table and retrieving is creating and updating.)(column 8, lines 2-4); updating the temporary query result data structure on the basis of the detected database change (i.e. *"Determining whether the criterion for notification has been satisfied and, if so, notifying the appropriate clients, is preferably accomplished by means of the Check Monitors procedure as illustrated in FIG. 3. As described above, the procedure is initiated (block 301) by a procedure call generated during an update session (the procedure may also be called directly by a client as will be discussed in more detail in a subsequent paragraph). Any update function call which might have resulted in a change to a monitored attribute value is detected (block 302). In most instances this accomplished by reference to the Function Change table and to a "Function Dependence" table (to be described hereafter)." "Because in this instance the procedure was called by a commit request made by an update client, the update client is the one which receives the notification. The Function Change table is cleared (block 307) and the procedure ends (block 308)." The preceding text clearly indicates that a Check Monitors procedure is a type of a trigger procedure that updates the temporary query data result, which is the Function Change table and further updates the client request, which is the initial query.)(column 8, lines 15-25 and 36-40); and if the query is re-executed against the database, returning the temporary query result data structure as query result to the issuing entity (i.e. *"Then the saved change notification is sent to any other client which had requested monitoring of the changed attribute value (block 209). If there is another update task to perform (block 210), such task is performed (block 203) and the above-described steps are repeated."* The preceding text clearly indicates that the saved change notification is the new results in the temporary query result data structure that updated the*

initial query request by the client and any other client which requested monitoring is the query re-execution.)(column 8, lines 49-54)..

As per claims 16 and 36, Risch teaches a method further comprising: receiving another query against the database; determining, whether the another query includes all query conditions defined by the query (i.e. *"Then the saved change notification is sent to any other client which had requested monitoring of the changed attribute value (block 209)."* The preceding text clearly indicates that any other client which had requested monitoring is another query against the database and the query condition is the monitoring of the changed attribute value.)(column 8, lines 49-52); and if so, determining another query result for the another query using the temporary query result data structure (i.e. *"The system creates the Function Change table (block 202) and then proceeds with an update task as directed by the client (block 203)."* The preceding text clearly illustrates that a temporary query result data structure is a function change table and the system is the re-execution of the query.)(column 8, lines 6-8).

As per claims 20 and 40, Risch teaches a method wherein the tracking of the changes to the database (i.e. *"The invoked procedure is sometimes referred to herein as the "tracking procedure" ("tracker" for short) because it "tracks"--that is, is invoked by a change in--the monitored attribute."*)(column 4, lines 1-3) is performed by a trigger procedure (i.e. *"Systems of "triggers" have been proposed for relational database systems; such triggers typically invoke a database procedure upon updates of user-specified base relations."*)(column 2, line 65; column 3, line 1).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 12 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Risch (U.S. Patent No. 5,471,629) in view of Beckman (U.S. Patent Pub. No. 2003/0182197).

As per claims 12 and 32, Risch does not explicitly teach teaches a method further comprising, after the determined timeframe is elapsed: determining, whether an extension time parameter can be identified; and if so, updating the timeframe on the basis of the identified extension time parameter.

Beckman teaches a method further comprising, after the determined timeframe is elapsed: determining, whether an extension time parameter can be identified; and if so, updating the timeframe on the basis of the identified extension time parameter (i.e. *"Position of the portable communication device is monitored 508 as described previously. Additionally, a determination is made as to whether the time parameter has been reached 510. If the time parameter has not been reached 510, the portable communication device may continue to determine with a location and a good are within predetermined proximity 512, and if so, notify the user 514. However, if the time parameter has been reached 510, the user may be notified 516, such as to enable the user to extend the time parameter, delete the request, and the like."* The preceding text clearly indicates that an extension

time parameter is applied when the user extends the time, thus updating the timeframe based on the extension time parameter.)(page 3, paragraph 29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Risch with the teachings of Beckman to include a method further comprising, after the determined timeframe is elapsed: determining, whether an extension time parameter can be identified; and if so, updating the timeframe on the basis of the identified extension time parameter with the motivation to improved efficiency of the database system and optimal results from each of a plurality of client programs. (Risch, column 5, lines 19-20).

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on 571-272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

FMS

Am Mofiz
Am Mofiz
Primary Examiner
TC 2100